# **Proyecto iteración 4**

Juliana Andrea Galeano Caicedo, William Felipe Mendez Ardila

Universidad de los Andes, Bogotá, Colombia

ja.galeanoc1@ uniandes.edu.co, w.mendez@uniandes.edu.co

Fecha de presentación: diciembre 6 de 2021

## **Análisis y modelo conceptual**

|  |  |
| --- | --- |
| Nombre | RF1 |
| Resumen | Permite registrar un usuario en el sistema, esta acción solo puede ser realizada por un usuario administrador de BancAndes para los empleados del banco (gerente general, gerentes de oficina y cajeros) y por un gerente de oficina para los clientes. |
| Entradas | |
| Credenciales: un login y una palabra clave. | |
| Rol: Por ejemplo, cliente, cajero, gerente de oficina, gerente general, entre otros. | |
| Datos de contacto: documento de identificación (tipo y número), nombre, nacionalidad, dirección física, dirección electrónica,  teléfono, ciudad, departamento y código postal. | |
| Resultados | |
| Guarda un nuevo usuario en la base de datos, no tiene ningún retorno. | |
| RNF asociados | |
| Persistencia | |

|  |  |
| --- | --- |
| Nombre | *RF2* |
| Resumen | Permite registrar un sitio físico donde el banco atiende a sus clientes, esta operación es realizada por el administrador de BancAndes. |
| Entradas | |
| Nombre de la oficina, dirección, el número de puntos de atención posibles y su gerente (debe ser un usuario ya registrado en el sistema), usuario que realiza la operación. | |
| Resultados | |
| Guarda una nueva oficina en base de datos, no tiene ningún retorno. | |
| RNF asociados | |
| Persistencia | |

|  |  |
| --- | --- |
| Nombre | *RF3* |
| Resumen | Permite registrar un punto de atención del banco, esta operación es realizada por el administrador de BancAndes. |
| Entradas | |
| Tipo de punto de atención, operaciones permitidas, cajeros, su localización y para los puestos de atención físicos se requiere también la oficina donde se encuentran, usuario que realiza la operación. | |
| Resultados | |
| Guarda un punto de atención en la base de datos, no tiene ningún retorno. | |
| RNF asociados | |
| Persistencia | |

|  |  |
| --- | --- |
| Nombre | *RF4* |
| Resumen | Permite al cliente abrir una cuenta en una oficina del banco, esta operación la realiza un gerente de oficina. |
| Entradas | |
| Tipo de cuenta, información del cliente, número de cuenta (se le debe asignar un número único) y usuario que realiza la operación. | |
| Resultados | |
| Guarda una cuenta en la base de datos, no tiene ningún retorno. | |
| RNF asociados | |
| Persistencia  Transaccionalidad: Debe ser transaccional para evitar dificultades si dos personas tratan de registrar el mismo cliente en dos oficinas. | |

|  |  |
| --- | --- |
| Nombre | *RF5* |
| Resumen | Permite cerrar una cuenta, lo cual significa que el saldo queda en cero y no se puede realizar operaciones sobre ella, sin embargo, aún se conservará toda la información de la cuenta, Esta operación la realiza el gerente de la oficina donde se creó la cuenta, a solicitud del cliente. |
| Entradas | |
| Usuario cliente, número de cuenta y usuario que realiza la operación. | |
| Resultados | |
| Modifica una cuenta dejándola con saldo cero e inactiva, no tiene ningún retorno. | |
| RNF asociados | |
| Persistencia  Transaccionalidad: Como cualquier operación que maneje dinero debe ser transaccional para que no se causen perdidas o desajustes para ningún involucrado. | |

|  |  |
| --- | --- |
| Nombre | *RF6* |
| Resumen | Permite registrar una operación sobre una cuenta y se actualiza el saldo, esta operación puede ser realizada por los cajeros y por los clientes. |
| Entradas | |
| Operación, número de cuenta, tipo de cuenta, usuario cliente o cajero. | |
| Resultados | |
| Guarda una operación sobre una cuenta y actualiza el saldo, no tiene ningún retorno. | |
| RNF asociados | |
| Persistencia  Transaccionalidad: Como cualquier operación que maneje dinero debe ser transaccional para que no se causen perdidas o desajustes para ningún involucrado. | |

|  |  |
| --- | --- |
| Nombre | *RF7* |
| Resumen | Permite registrar un préstamo que ya fue aprobado, esta operación la realiza un usuario gerente de una oficina. |
| Entradas | |
| Número de cuenta, monto del préstamo, el interés, el número de cuotas, el día del mes en el cual se debe pagar la cuota, el valor de la cuota mínima, el cliente al que se le aprobó el préstamo, usuario que realiza la operación. | |
| Resultados | |
| Guarda un préstamo en la base de datos, no tiene ningún retorno. | |
| RNF asociados | |
| Persistencia | |

|  |  |
| --- | --- |
| Nombre | *RF8* |
| Resumen | Permite registrar una operación sobre un préstamo y actualiza el saldo pendiente. Esta operación puede ser realizada por los cajeros. |
| Entradas | |
| Número de cuenta, el cliente dueño del préstamo, valor del pago (debe ser o igual a la cuota mínima definida para el préstamo o mayor, pero en ningún caso de debe poder realizar un pago menor a la cuota establecida), usuario cajero. | |
| Resultados | |
| Guarda una operación sobre un préstamo en la base de datos y actualiza el saldo, no tiene ningún retorno. | |
| RNF asociados | |
| Persistencia  Transaccionalidad: Como cualquier operación que maneje dinero debe ser transaccional para que no se causen perdidas o desajustes para ningún involucrado. | |

|  |  |
| --- | --- |
| Nombre | *RF9* |
| Resumen | Esta operación se realiza cuando el cliente ha pagado toda su deuda (el saldo pendiente del préstamo es cero). Por reglamentación bancaria la información del préstamo debe persistir. Esta operación la hace el gerente de oficina. |
| Entradas | |
| Número de cuenta, cliente dueño del préstamo, usuario que realiza la operación. | |
| Resultados | |
| Actualiza el saldo del préstamo dejando el saldo pendiente en cero, no tiene ningún retorno. | |
| RNF asociados | |
| Persistencia  Transaccionalidad: Como cualquier operación que maneje dinero debe ser transaccional para que no se causen perdidas o desajustes para ningún involucrado. | |

|  |  |
| --- | --- |
| Nombre | *RFC1* |
| Resumen | Permite consultar cuentas en BancAndes. Este requerimiento tiene limitaciones de información dependiendo del tipo de usuario que realice la consulta (cuando un cliente hace esta consulta, la respuesta sólo contiene la información de sus cuentas, un gerente de oficina obtiene la información de las cuentas de su oficina y el gerente general obtiene toda la información). |
| Entradas | |
| Usuario que realiza la consulta. | |
| Resultados | |
| Cuentas agrupadas y ordenadas dependiendo de los intereses del usuario que realizó la consulta. | |
| RNF asociados | |
| Privacidad y concurrencia. | |

|  |  |
| --- | --- |
| Nombre | *RFC2* |
| Resumen | Permite consultar la información de un cliente de BancAndes. Este requerimiento tiene limitaciones de información dependiendo del tipo de usuario que realice la consulta (cuando un cliente hace esta consulta, la respuesta sólo contiene la información de sus cuentas, un gerente de oficina obtiene la información de las cuentas de su oficina y el gerente general obtiene toda la información). |
| Entradas | |
| Usuario que realiza la consulta, ID del cliente del que desea conocer la información. | |
| Resultados | |
| Toda la información del del cliente buscado incluyendo su tipo, su nombre, las cuentas que tiene en el banco, la(s) oficina(s) donde tiene cuentas, los préstamos, las operaciones que ha realizado, y demás. Los resultados pueden ser filtrados por tipo de cuenta, cuenta, saldo, operaciones en un rango de fechas y valor. Se ordenará y agrupará la información según los intereses del usuario que realizó la consulta. | |
| RNF asociados | |
| Privacidad y concurrencia. | |

|  |  |
| --- | --- |
| Nombre | *RFC3* |
| Resumen | Permite consultar los 10 tipos de operaciones de mayor movimiento en el sistema por oficina, en un rango de fechas. Esta operación puede ser realizada por los gerentes de oficina y el gerente general, sin embargo, tiene limitaciones de información dependiendo del tipo de usuario que realice la consulta (cuando un gerente de oficina hace esta consulta obtiene la información de las operaciones sobre las cuentas de su oficina y el gerente general obtiene toda la información). |
| Entradas | |
| Usuario que realiza la consulta y rango de fechas. | |
| Resultados | |
| La información del tipo de operación, el punto de atención, el valor promedio, la cantidad de veces que ha sido realizada. El resultado debe estar ordenado por el número de veces realizada. | |
| RNF asociados | |
| Privacidad y concurrencia. | |

|  |  |
| --- | --- |
| Nombre | *RFC3* |
| Resumen | Permite consultar el usuario que ha realizado mayor cantidad de operaciones de cierto tipo o por valor mayor a una cantidad especifica. Esta operación puede ser realizada por los gerentes de oficina y el gerente general, sin embargo, tiene limitaciones de información dependiendo del tipo de usuario que realice la consulta (cuando un gerente de oficina hace esta consulta obtiene la información de las operaciones sobre las cuentas de su oficina y el gerente general obtiene toda la información). |
| Entradas | |
| Usuario que realiza la consulta, tipo de operación o cantidad especifica. | |
| Resultados | |
| Información de el o los usuarios que cumplan con las condiciones dadas. | |
| RNF asociados | |
| Privacidad y concurrencia. | |

|  |  |
| --- | --- |
| Nombre | *RFC7* |
| Resumen | Permite mostrar las operaciones bancarias en un rango de tiempo (dado por el usuario), que correspondan a un criterio de búsqueda asociado con la operación realizada (tipo de transacción, monto, …), también dado por el usuario. Esta operación solo es realizada por el gerente general. |
| Entradas | |
| Rango de tiempo y criterio de búsqueda. | |
| Resultados | |
| Una lista con todas las operaciones bancarias que cumplan con las condiciones. | |
| RNF asociados | |
| Privacidad, concurrencia y eficiencia en las consultas. | |

|  |  |
| --- | --- |
| Nombre | *RFC8* |
| Resumen | Permite mostrar las operaciones bancarias en un rango de tiempo (dado por el usuario), que no correspondan a un criterio de búsqueda asociado con la operación realizada (tipo de transacción, monto, …), también dado por el usuario. Esta operación solo es realizada por el gerente general. |
| Entradas | |
| Rango de tiempo y criterio de búsqueda. | |
| Resultados | |
| Una lista con todas las operaciones bancarias que cumplan con las condiciones. | |
| RNF asociados | |
| Privacidad, concurrencia y eficiencia en las consultas. | |

|  |  |
| --- | --- |
| Nombre | *RFC9* |
| Resumen | Permite mostrar todas las consignaciones con monto mayor a un valor X realizadas por clientes que tienen productos de un tipo Y. Los valores de X y Y son dados por el usuario. Esta operación solo es realizada por el gerente general. |
| Entradas | |
| Valor (X) y productos (Y). | |
| Resultados | |
| Una lista con todas las operaciones de tipo consignación que cumplan con las condiciones. | |
| RNF asociados | |
| Privacidad, concurrencia y eficiencia en las consultas. | |

|  |  |
| --- | --- |
| Nombre | *RFC10* |
| Resumen | Permite, dado el identificador de dos puntos de atención, mostrar la información de los clientes que han realizado operaciones en esos dos puntos de atención, incluyendo también la información de las operaciones realizadas. Esta operación solo es realizada por el gerente general. |
| Entradas | |
| Identificador de dos puntos de atención. | |
| Resultados | |
| Una lista con todos los clientes y las operaciones que realizaron en estos puntos de atención. | |
| RNF asociados | |
| Privacidad, concurrencia y eficiencia en las consultas. | |

## **Diagrama UML**

Diagrama

Descripción generada automáticamente

## **Diseño de la base de datos**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Oficina |  |  |  |  |  |
| idOficina | nombre | dirección | numPAtención | tipoIdGerente | idGerente |
| PK, SA, NC | NN, ND, NC, UA | NN, ND, NC, UA | NN,DD | FK(idOficina, tipoIdGerente) | FK(idOficina, idGerente) |
|  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PuestoAtencion |  |  |  |  |  |
| idPAtencion | tipoPuesto | localizacion | tipoIdCajero | idCajero | Oficina |
| PK, SA, NC | NN, NC, CK(Oficina, ATM, Web, Movil), UA | NN, NC,SA | FK(idOficina, tipoIdCajero) | FK(idOficina, idCajero) | FK(idPAtencion, idOficina) |
|  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Producto |  |  |  |  |  |  |  |
| idProducto | estado | valorActual | fechaCreacion | fechaUltimoMov | Oficina | tipoIdUsuario | idUsuario |
| PK, SA, NC | NN, CK(Activa, Cerrada, Desactivada) | NN, UA, DD, CK ( >= 0) | NN, NC, SA | NULL, SA, DD | NN,FK(idProducto, idOficina) | NN, FK(idProducto, tipoIdUsuario) | NN, FK(idProducto, idUsuario) |
|  |  |  |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Cuenta Juridica |  |  |  |
| idProducto | valorAPagar | valorAPagar | CuentaPagar |
| PK, SA, NC | NN, UA | NN, UA | NN, FK(idProducto, idProducto) |
|  |  |  |  |

|  |  |
| --- | --- |
| Cuenta |  |
| idProducto | TipoCuenta |
| PK, SA, NC | NN, CK(Ahorros, Corriente) |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| AFC |  |  |  |
| idProducto | fechaPactada | dineroInicial | interes |
| PK, SA, NC | NN,UA | NN, NC, UA | NN, NC, UA |
|  |  |  |  |
| CDT |  |
| idProducto | fechaPactada |
| PK, SA, NC | NN,UA |
|  |  |

|  |  |  |
| --- | --- | --- |
| Préstamo |  |  |
| idProducto | fechaPactada | cuotaMinima |
| PK, SA, NC | NN,UA | NN, NC, UA |
|  |  |  |

|  |  |
| --- | --- |
| CuentaJuridica\_Cuenta |  |
| IdCuentaJuridica | IdCuenta |
| NN, FK(idProducto, idProducto) | NN, FK(idProducto, idProducto) |
|  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Accion** |  |  |  |  |  |
| idAccion | compania | valorAccion | Producto | tipoIdUsuario | idUsuario |
| PK, SA, NC | NN, UA | UA | NN, FK(idAccion, idProducto) | NN, FK(idAccion, tipoIdUsuario) | NN, FK(idAccion, idUsuario) |
|  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Operación** | |  |  | |  | |  | | |  |
| idOperacion | | tipoOperacion | valor | | fecha | | PuestoAtencion | | | tipoIdUsuario |
| PK, SA, NC | | NN, NC, UA | NN, NC,SA | | NN, NC,SA | | NN, FK(idOperacion, idPAtencion ) | | | NN, FK(idOperacion, tipoIdUsuario) |
|  | |  |  | |  | |  | | |  |
| idUsuario | Cuenta\_origen | | | Cuenta\_destino | | tipoIdEmpleado | | idEmpleado |
| NN, FK(idOperacion, idUsuario) | NN, FK(idOperacion, idProducto) | | | NN, FK(idOperacion, idProducto) | | FK(idOperacion, tipoIdEmpleado) | | FK(idOperacion, idEmpleado) |
|  |  | | |  | |  | |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Usuario** |  |  | |  | |  | |  | |
| Id | tipoId | tipoUsuario | | login | | clave | | nombre | |
| PK, UA | PK, UA, CK (CI, CC, TP, CE) | NN, NC, CK(Natural, Juridica), UA | | NN, ND, NC, UA | | NN, UA | | NN, UA | |
|  |  |  | |  | |  | |  | |
| nacionalidad | dirFisica | | dirElectronica | | telefono | | ciudad | departamento | codigoPostal | |
| NN, UA | NN, UA | | NN, UA, ND | | NN, UA | | NN, UA | NN,UA | NN, UA | |
|  |  | |  | |  | |  |  |  | |

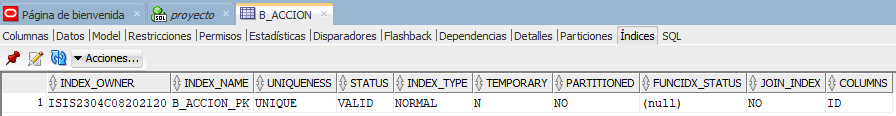
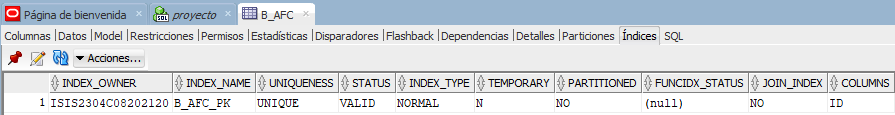
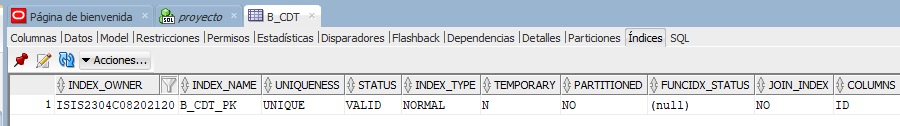
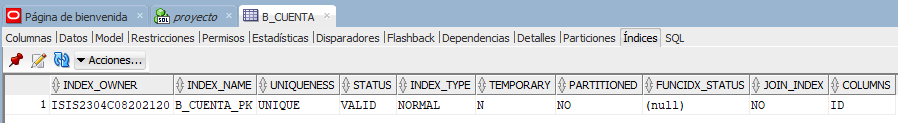
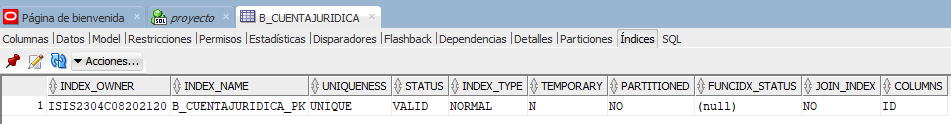
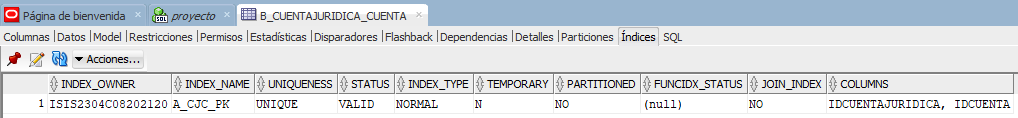
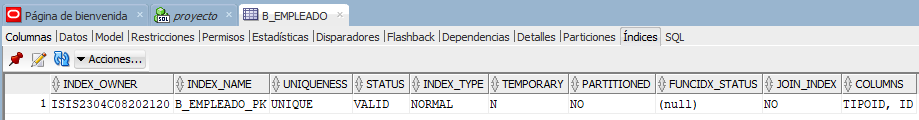
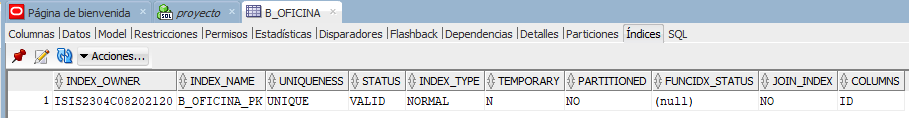
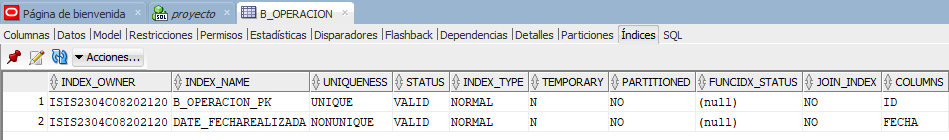
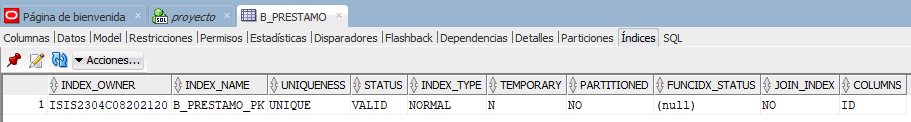
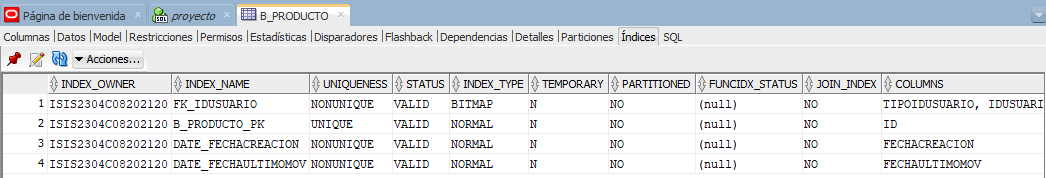
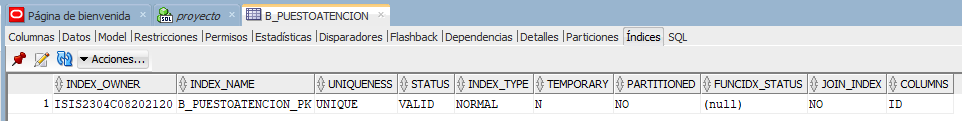
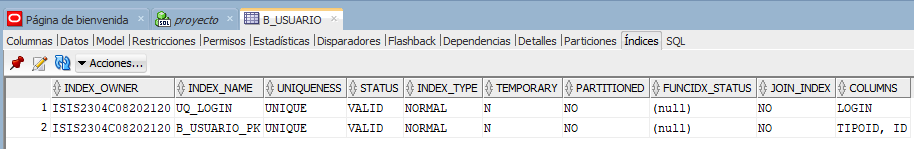
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Empleado** |  |  |  |  |
| Id | tipoId | tipoEmpleado | Oficina | PuestoAtencion |
| PK, UA | PK, UA, CK (CI, CC, TP, CE) | NN, CK(Gerente general, Administrador, Gerente oficina, Cajero), UA | FK(numId, idOficina) | FK(NumId, idPAtencion ) |
|  |  |  |  |  |

## **Diseño de la aplicación**

## Debido a que en el modelo ya se estaban almacenando datos relevantes para las operaciones y los puntos de atención, en esta iteración no se realizaron cambios en el diseño más allá de la capitalización de las palabras presentes en las enumeraciones y una corrección a un error del diagrama del modelo en la clase Usuario que contenía una lista de String en la dirección cuando en realidad es solo un String.

## **Diseño físico**

Los índices existentes en las tablas son los siguientes:



Los índices que Oracle generó automáticamente fueron todos los existentes en las llaves primarias de las tablas y el generado sobre la columna login de la tabla usuario, pues esta columna es una llave candidata al ser única para cada tupla.

Por otro lado, los índices creados por el grupo fueron Date\_fechaRealizada en B\_operación, Date\_fechaCreacion en B\_producto, Date\_fechaUltimoMov en B\_producto y FK\_idUsuario ON b\_producto. Los tres primeros son índices B+ secundarios simples sobre columnas de fechas en las tablas pues este es un parámetro para filtrar común en las consultas requeridas. Por otro lado, el último indice creado es un índice Hash secundario compuesto por las columnas tipoIdUsuario e idUsuario, las cuales son una referencia a la tabla B\_usuario y son un criterio de Join común para los requerimientos de consulta presentados.

**RFC7:**

Para este requerimiento se siguen las siguientes consultas SQL:

S*ELECT \* FROM B\_operacion WHERE fecha >= fechainferior AND fecha <= fechasuperior AND tipooperacion = tipoOperacion*

Con los parámetros fechainferior (En formato TimeStamp), fechasuperior (En formato TimeStamp) y tipoOperacion para filtrar por un rango de fechas y un tipo de operación.

*SELECT \* FROM B\_operacion WHERE fecha >= fechainferior AND fecha <= fechasuperior AND valor >= valorInferior AND valor <= valorSuperior*

Con los parámetros fechainferior (En formato TimeStamp), fechasuperior (En formato TimeStamp), valorInferior y valorSuperior para filtrar por un rango de fechas y un rango de valores.

*SELECT \* FROM B\_operacion WHERE fecha >= fechainferior AND fecha <= fechasuperior AND id >= idInferior AND id <= idSuperior*

Con los parámetros fechainferior (En formato TimeStamp), fechasuperior (En formato TimeStamp), idInferior e idSuperior para filtrar por un rango de fechas y un rango de id de la operación.

*SELECT \* FROM B\_operacion WHERE fecha >= fechainferior AND fecha <= fechasuperior AND idpuestoatencion = idPuestoAtencion*

Con los parámetros fechainferior (En formato TimeStamp), fechasuperior (En formato TimeStamp) e idPuestoAtencion para filtrar por un rango de fechas y operaciones realizadas en el puesto de atención con esa id.

*SELECT \* FROM B\_operacion WHERE fecha >= fechainferior AND fecha <= fechasuperior AND cuenta\_origen = cuenta\_origen*

Con los parámetros fechainferior (En formato TimeStamp), fechasuperior (En formato TimeStamp) y cuenta\_origen para filtrar por un rango de fechas y una cuenta de origen específica.

*SELECT \* FROM B\_operacion WHERE fecha >= fechainferior AND fecha <= fechasuperior AND cuenta\_destino = cuenta\_destino*

Con los parámetros fechainferior (En formato TimeStamp), fechasuperior (En formato TimeStamp) y cuenta\_destino para filtrar por un rango de fechas y una cuenta de destino específica.

*SELECT \* FROM B\_operacion WHERE fecha >= fechainferior AND fecha <= fechasuperior AND tipoidusuario = tipoIdUsuario AND idusuario = idUsuario*

Con los parámetros fechainferior (En formato TimeStamp), fechasuperior (En formato TimeStamp), tipoIdUsuario e idUsuario para filtrar por un rango de fechas y un cliente específico.

*SELECT \* FROM B\_operacion WHERE fecha >= fechainferior AND fecha <= fechasuperior AND tipoidempleado = tipoIdEmpleado AND idempleado = idEmpleado*

Con los parámetros fechainferior (En formato TimeStamp), fechasuperior (En formato TimeStamp), tipoIdEmlpeado e idEmpleado para filtrar por un rango de fechas y un cajero específico.

La relación de los parámetros de entrada con los resultados obtenidos es:

* Captura de pantalla de computadora

  Descripción generada automáticamenteCaptura de pantalla de computadora

  Descripción generada automáticamentefechainferior y fechasuperior: Entre más grande sea este rango, mayor será la cantidad de datos.

Ilustración 1 Sentencia con un rango grande

Ilustración 2 Sentencia con un rango pequeño

* Interfaz de usuario gráfica, Texto, Aplicación

  Descripción generada automáticamenteCaptura de pantalla de computadora

  Descripción generada automáticamentetipoOperacion: Cambia el número de resultados dependiendo de la distribución de los datos.

Ilustración 3 Sentencia con tipoOperacion Transferencia

Ilustración 4 Sentencia con tipoOperacion Retiro

* Interfaz de usuario gráfica, Texto, Aplicación

  Descripción generada automáticamenteCaptura de pantalla de computadora

  Descripción generada automáticamentevalorInferior y valorSuperior: Entre más grande sea este rango, mayor será la cantidad de datos.

Ilustración 5 Sentencia con un rango grande

Ilustración 6 Sentencia con un rango pequeño

* Una captura de pantalla de una computadora

  Descripción generada automáticamenteCaptura de pantalla de computadora

  Descripción generada automáticamenteidInferior y idSuperior: Entre más grande sea este rango, mayor será la cantidad de datos.

Ilustración 7 Sentencia con un rango grande

Ilustración 8 Sentencia con un rango pequeño

* Captura de pantalla de computadora

  Descripción generada automáticamenteCaptura de pantalla de computadora

  Descripción generada automáticamenteidPuestoAtencion: Cambia el número de resultados dependiendo de la distribución de los datos.

Ilustración 9 Sentencia con idPuestoAtencion 434

Ilustración 10 Sentencia con idPuestoAtencion 486

* Captura de pantalla de computadora

  Descripción generada automáticamenteCaptura de pantalla de computadora

  Descripción generada automáticamentecuenta\_origen: Cambia el número de resultados dependiendo de la distribución de los datos.

Ilustración 11 Sentencia con cuenta\_origen 67

Ilustración 12 Sentencia con cuenta\_origen 512

* cuenta\_destino: Cambia el número de resultados dependiendo de la distribución de los datos que cumplen la condición.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteCaptura de pantalla de computadora

Descripción generada automáticamente

Ilustración 13 Sentencia con cuenta\_destino 53

Ilustración 14 Sentencia con cuenta\_destino 23

* Captura de pantalla de computadora

  Descripción generada automáticamenteCaptura de pantalla de computadora

  Descripción generada automáticamentetipoIdUsuario e idUsuario: Cambia el número de resultados dependiendo de la distribución de los datos.

Ilustración 15 Sentencia con id CC 75

Ilustración 16 Sentencia con id CI 3

* Captura de pantalla de computadora

  Descripción generada automáticamenteCaptura de pantalla de computadora

  Descripción generada automáticamentetipoIdEmpleado e idEmpleado: Cambia el número de resultados dependiendo de la distribución de los datos.

Ilustración 17 Sentencia con el id CE 691

Ilustración 18 Sentencia con id TP 8

Los valores de entrada que cambian el plan de ejecución son:

* El rango de fechas y el rango de ids: Cuando se decide filtrar por estos valores el procesador se encarga de usar los índices existentes en estas columnas. Sin embargo, cuando alguno de estos dos rangos contiene la tabla completa esta optimización no es válida y se recorre entera. Para el resto de los parámetros no se generan cambios en el plan pues cada uno hace que se necesite una comparación adicional y siempre se recorra la tabla entera.

Las selectividades se validan por:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente con confianza media

Imagen que contiene Texto

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Imagen que contiene Texto

Descripción generada automáticamente

Imagen que contiene Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza baja

Con una consulta que incluye todas las fechas se puede verificar que no es viable poner índices en el tipo de operación ya que para cada uno de los tipos la selectividad mínima es del 31 %. Sn embargo, parece viable poner índices en valor, idpuestoatencion, cuenta\_origen, cuenta\_destino, usuario y empleado, ya que sus selectividades en datos de prueba aleatorios son muy pequeñas.

El plan de ejecución pensado por el grupo fue:

Para la mayoría de los filtros el plan de ejecución deberá recorrer la mayoría de la tabla para poder realizar las comparaciones y filtrar, para los que no, será un recorrido con los índices de la fecha y la llave primaria.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteEl plan de ejecución que muestra SQL Developer para la mayoría de las consultas es:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteEl plan que se muestra cuando se filtra por id y por fecha es:

Con estos planes podemos confirmar que la intuición del grupo es válida.

**RFC8:**

Para este requerimiento se sigue(n) la(s) siguiente(s) consulta(s) SQL:

*SELECT \* FROM B\_operacion WHERE fecha >= fechainferior AND fecha <= fechasuperior AND tipooperacion <> tipoOperacion*

Con los parámetros fechainferior (En formato TimeStamp), fechasuperior (En formato TimeStamp) y tipoOperacion para filtrar por un rango de fechas y un tipo de operación.

*SELECT \* FROM B\_operacion WHERE fecha >= fechainferior AND fecha <= fechasuperior AND NOT (valor < valorInferior OR valor > valorSuperior)*

Con los parámetros fechainferior (En formato TimeStamp), fechasuperior (En formato TimeStamp), valorInferior y valorSuperior para filtrar por un rango de fechas y un rango de valores.

*SELECT \* FROM B\_operacion WHERE fecha >= fechainferior AND fecha <= fechasuperior AND (id < idInferior OR id > idSuperior)*

Con los parámetros fechainferior (En formato TimeStamp), fechasuperior (En formato TimeStamp), idInferior e idSuperior para filtrar por un rango de fechas y un rango de id de la operación.

*SELECT \* FROM B\_operacion WHERE fecha >= fechainferior AND fecha <= fechasuperior AND idpuestoatencion <> idPuestoAtencion*

Con los parámetros fechainferior (En formato TimeStamp), fechasuperior (En formato TimeStamp) e idPuestoAtencion para filtrar por un rango de fechas y operaciones realizadas en el puesto de atención con esa id.

*SELECT \* FROM B\_operacion WHERE fecha >= fechainferior AND fecha <= fechasuperior AND cuenta\_origen <> cuenta\_origen*

Con los parámetros fechainferior (En formato TimeStamp), fechasuperior (En formato TimeStamp) y cuenta\_origen para filtrar por un rango de fechas y una cuenta de origen específica.

*SELECT \* FROM B\_operacion WHERE fecha >= fechainferior AND fecha <= fechasuperior AND cuenta\_destino <> cuenta\_destino*

Con los parámetros fechainferior (En formato TimeStamp), fechasuperior (En formato TimeStamp) y cuenta\_destino para filtrar por un rango de fechas y una cuenta de destino específica.

*SELECT \* FROM B\_operacion WHERE fecha >= fechainferior AND fecha <= fechasuperior AND NOT (tipoidusuario = tipoIdUsuario AND idusuario = idUsuario)*

Con los parámetros fechainferior (En formato TimeStamp), fechasuperior (En formato TimeStamp), tipoIdUsuario e idUsuario para filtrar por un rango de fechas y un cliente específico.

*SELECT \* FROM B\_operacion WHERE fecha >= fechainferior AND fecha <= fechasuperior AND NOT (tipoidempleado = tipoIdEmpleado AND idempleado = idEmpleado)*

Con los parámetros fechainferior (En formato TimeStamp), fechasuperior (En formato TimeStamp), tipoIdEmlpeado e idEmpleado para filtrar por un rango de fechas y un cajero específico.

La relación de los parámetros de entrada con los resultados obtenidos es:

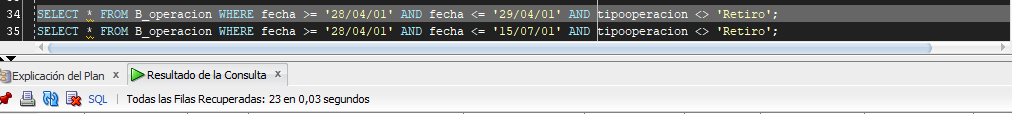
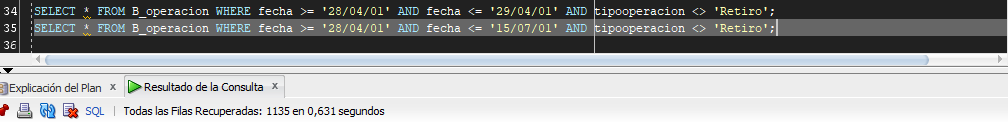
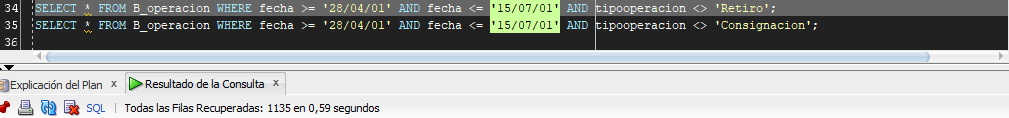
* fechainferior y fechasuperior: Entre más grande sea este rango, menor será la cantidad de datos.

Ilustración 19 Sentencia con un rango mayor a un mes

Ilustración 18 Sentencia con un rango de un día

* Interfaz de usuario gráfica, Texto, Aplicación

  Descripción generada automáticamentetipoOperacion: Cambia el número de resultados dependiendo de la distribución de los datos.

Ilustración 21 Sentencia con tipooperacion Consignacion

Ilustración 19 Sentencia con tipooperacion Retiro

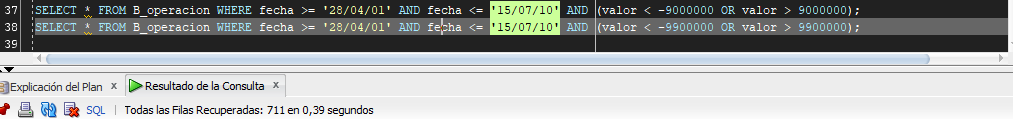
* valorInferior y valorSuperior: Entre más grande sea este rango, mayor será la cantidad de datos.

Ilustración 20 Sentencia con rango -9900000 a 9900000

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 21 Sentencia con rango -9000000 a 9000000

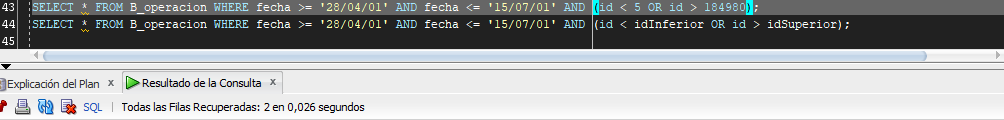
* idInferior y idSuperior: Entre más grande sea este rango, mayor será la cantidad de datos.

Ilustración 25 Sentencia con un rango de 5 a 184980

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Ilustración 24 Sentencia con un rango de 50000 a 184000

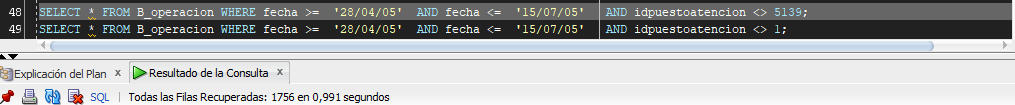
* idPuestoAtencion: Cambia el número de resultados dependiendo de la distribución de los datos.

Ilustración 22 Sentencia con idpuestoatencion 5139

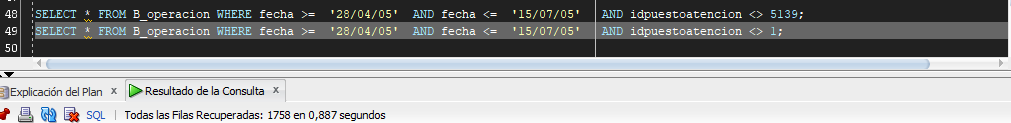


Ilustración 23 Sentencia con idpuestoatencion 1

* cuenta\_origen: Cambia el número de resultados dependiendo de la distribución de los datos.

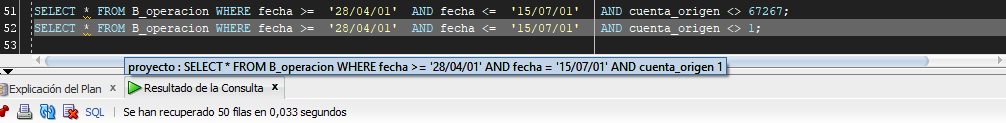
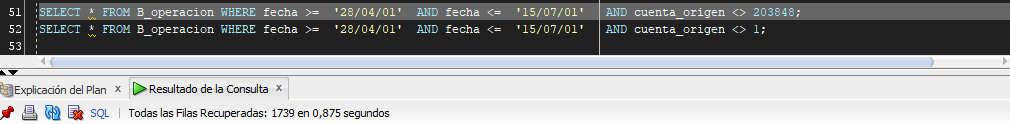


Ilustración 24 Sentencia con cuenta\_origen 203848

Ilustración 25 Sentencia con cuenta\_origen 1

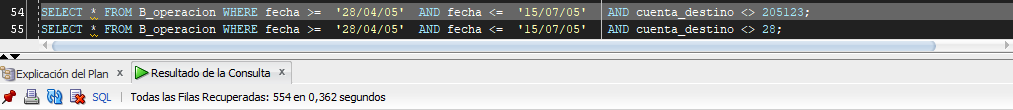
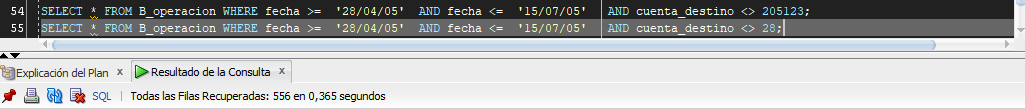
* cuenta\_destino: Cambia el número de resultados dependiendo de la distribución de los datos que cumplen la condición.

Ilustración 26 Sentencia con cuenta\_destino 28

Ilustración 27 Sentencia con cuenta\_destino 205123

* Interfaz de usuario gráfica, Texto, Aplicación, Sitio web

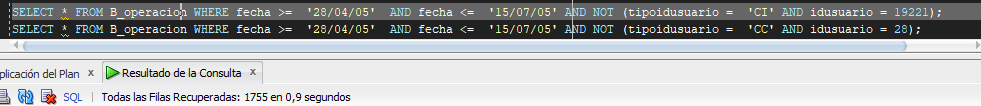
  Descripción generada automáticamentetipoIdUsuario e idUsuario: Cambia el número de resultados dependiendo de la distribución de los datos.

Ilustración 28 Sentencia con id CC 28

Ilustración 29 Sentencia con id CI 19221

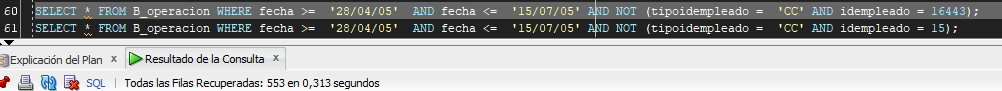
* tipoIdEmpleado e idEmpleado: Cambia el número de resultados dependiendo de la distribución de los datos. 

Ilustración 30 Sentencia con idEmpleado CC 16443

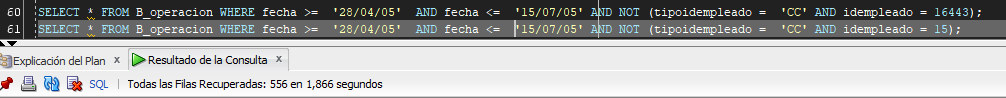


Ilustración 31 Sentencia con idEmpleado CC 15

Los valores de entrada que cambian el plan de ejecución son:

* El rango de fechas y el rango de ids: Cuando se decide filtrar por estos valores el procesador se encarga de usar los índices existentes en estas columnas. Sin embargo, cuando alguno de estos dos rangos contiene la tabla completa esta optimización no es válida y se recorre entera. Para el resto de los parámetros no se generan cambios en el plan pues cada uno hace que se necesite una comparación adicional y siempre se recorra la tabla entera.

Las selectividades se validan por:

Interfaz de usuario gráfica, Patrón de fondo

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza media

Texto

Descripción generada automáticamente con confianza baja

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

Con una consulta que incluye todas las fechas se puede verificar que no es viable poner índices en ninguno de los atributos relacionados con los criterios ya que para esta consulta las selectividades suelen ser muy altas.

El plan de ejecución pensado por el grupo fue:

Del mismo que en el requerimiento anterior, este deberá recorrer las tablas de manera entera para la mayoría de los filtros a excepción de cuando se filtra por id, pues se puede recorrer parcialmente por los índices existentes.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteLos planes de ejecución que muestra SQL Developer para los filtros de idEmpleado, idCliente, cuenta\_destino, cuenta\_origen, idPuestoAtencion, valor, tipoOperación y un rango de fechas que contiene toda la tabla con variaciones en el cuerpo de lógica son:

Cuando se filtra por un rango de id que no es muy reducido y fechas en un rango de un día los planes son respectivamente:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Luego de ver los planes de SQL Developer podemos ver que la intuición del grupo falló en el caso en el que se filtra por tipo de operación y en ese caso se accede por un índice en la comparación tipoOperacion <> ‘Retiro’.

**RFC9:**

Para este requerimiento se sigue(n) la(s) siguiente(s) consulta(s) SQL:

* SELECT o.\*

*FROM b\_usuario u JOIN b\_producto p ON (u.tipoid = p.tipoIdUsuario AND u.id = p.idUsuario) JOIN b\_prestamo c ON p.id = c.id JOIN b\_operacion o ON (o.tipoidUsuario = u.tipoid AND o.idusuario = u.id)*

*WHERE valor > X AND o.tipooperacion = 'Consignacion';*

* *SELECT o.\**

*FROM b\_usuario u JOIN b\_producto p ON (u.tipoid = p.tipoIdUsuario AND u.id = p.idUsuario) JOIN b\_cuenta c ON p.id = c.id JOIN b\_operacion o ON (o.tipoidUsuario = u.tipoid AND o.idusuario = u.id)*

*WHERE valor > X AND o.tipooperacion = 'Consignacion';*

* *SELECT o.\**

*FROM b\_usuario u JOIN b\_producto p ON (u.tipoid = p.tipoIdUsuario AND u.id = p.idUsuario) JOIN b\_cuentajuridica c ON p.id = c.id JOIN b\_operacion o ON (o.tipoidUsuario = u.tipoid AND o.idusuario = u.id)*

*WHERE valor > X AND o.tipooperacion = 'Consignacion';*

Con el parámetro X para filtrar las consignaciones con un valor mayor a X. En este requerimiento el filtro del tipo de producto Y está implementado en la interfaz y dependiendo de este se llama a una de las 3 sentencias listadas arriba.

La relación de los parámetros de entrada con los resultados obtenidos es:

* Interfaz de usuario gráfica

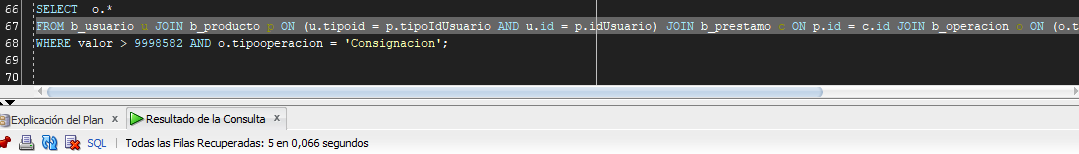
  Descripción generada automáticamenteX: Cuando este valor es menor implica que se filtrarán menos resultados y que los datos obtenidos van a aumentar. 

Ilustración 32 Sentencia con un valor X de 9800000

Ilustración 33 Sentencia con un valor x de 9998582

* Captura de pantalla de computadora

  Descripción generada automáticamenteCaptura de pantalla de computadora

  Descripción generada automáticamenteEl tipo de producto escogido: Cuando se cambia este parámetro los datos obtenidos van a variar según la distribución de los datos.

Ilustración 34 Sentencia con Y Cuenta

Ilustración 35 Sentencia con Y Prestamo

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteLos valores de entrada que cambian el plan de ejecución son:

Ilustración 36 Secuencia con Y Cuenta Jurídica

* Ninguno: Al ser necesario revisar datos de las tablas b\_producto, b\_operacion, usuario, b\_cuenta, b\_prestamo y b\_cuentaJuridica el plan de ejecución siempre se mantiene constante en recorridos enteros de las tablas.

Las selectividades se validan por:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Al realizar una consulta con datos aleatorios se puede considerar que no es viable implementar índices en los atributos involucrados en esta consulta ya que en esta consulta las selectividades son altas.

El plan de ejecución pensado por el grupo fue:

Se planteo que el plan necesitará un recorrido total por la tabla de operaciones para realizar la comparación con el parámetro X, además, dependiendo del parámetro Y, se deberá recorrer la tabla del producto Y para asociarla a su usuario luego de acceder a ella por medio del indice en producto que referencia a usuario.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteLos planes de ejecución que muestra SQL Developer para filtrar por una id de operación alta, prestamos, cuentas y cuentas jurídicas respectivamente son:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Los planes nos muestran que para obtener la información el plan en realidad recorre 2 veces la tabla producto, una indexada para relacionarse con la tabla de usuarios y una para relacionarse con la tabla Y.

**RFC10:**

Para este requerimiento se sigue(n) la(s) siguiente(s) consulta(s) SQL:

*SELECT u.\* FROM b\_usuario u JOIN b\_operacion o ON o.tipoidusuario = u.tipoid AND o.idusuario = u.id WHERE o.idpuestoatencion = id1 OR o.idpuestoatencion = id2 ORDER BY u.id;*

*SELECT o.\* FROM b\_usuario u JOIN b\_operacion o ON o.tipoidusuario = u.tipoid AND o.idusuario = u.id WHERE o.idpuestoatencion = id1 OR o.idpuestoatencion = id2 ORDER BY u.id;*

Con los parámetros id1 e id2 que permiten filtrar por los puntos de atención con dichas ids.

La relación de los parámetros de entrada con los resultados obtenidos es:

* Imagen que contiene Interfaz de usuario gráfica

  Descripción generada automáticamenteCaptura de pantalla de computadora

  Descripción generada automáticamenteid1 e id2: Cuando se cambia este parámetro los datos obtenidos van a variar según la distribución de los datos.

Ilustración 37 Sentencia que trae los datos de las operaciones

Ilustración 38 Sentencia que trae los datos de los usuarios

Los valores de entrada que cambian el plan de ejecución son:

* id1 e id2: El único cambio que puede generarse en el plan de ejecución con estos parámetros es asignar el mismo valor en ambos, esto va a ahorrar una comparación.

Las selectividades se validan por:

Texto

Descripción generada automáticamente

Patrón de fondo

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

Texto

Descripción generada automáticamente con confianza baja

Al realizar una consulta con datos aleatorios se puede considerar que es viable implementar índices en el id del puesto de atención ya que las selectividades en esta consulta suelen ser .

El plan de ejecución pensado por el grupo fue:

El plan va a necesitar que se recorra de forma completa la tabla de operaciones para realizar la comparación referente a las id1 e id2, luego de esto va a ser necesario acceder a la tabla de usuarios para tomar su información. Para efectos prácticos en este requerimiento se realizó en dos consultas para unificarlas en el programa Java, por lo que estos planes se basarían en acceder a la tabla de operaciones y relacionarla con la tabla de usuarios mientras se filtran los puntos de atención para obtener los datos del usuario, recorrer la tabla de operaciones para filtrar id1 e id2 y luego de eso en el programa unificar los datos.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteLos planes de ejecución que muestra SQL Developer para obtener los datos de usuarios y operaciones son:

## **Construcción de la aplicación, ejecución de pruebas y análisis de resultados**

Para la generación de datos se hizo uso de un script de Python que se encargaba de generar sentencias de inserts con datos aleatorios para una tabla en un archivo SQL que después se ejecutaba en SQL Developer. La lógica del script se basa en:

*Contenedor de datos necesarios para crear tabla1*

*Contenedor de datos necesarios para crear tabla2*

*tabla1():*

*Abrir archivo para escribir*

*Para cada número en un rango:*

*Generar sentencia "Insert into tabla1 values ("*

*Añadir a la sentencia su id*

*Añadir dato aleatorio 1*

*Añadir dato aleatorio 2*

*...*

*Añadir dato aleatorio n*

*Añadir datos dependientes de otras tablas*

*Almacenar datos que necesita tabla2*

*Terminar la sentencia con ");\n"*

*Escribir la sentencia en tabla1.sql*

*Cerrar el archivo tabla1.sql*

*Escribir el número de sentencias generadas*

*Registrar el tiempo usado para generar el archivo*

Este programa se adaptó para cada tabla y se insertaron datos en los rangos suficientes para generar 1’081.555 sentencias SQL y estas se unificaron en un archivo completo llamado bancandes.sql, el cual cuenta las presenta en un orden válido para las ejecuciones sin errores generados por restricciones de integridad relacionadas con llaves foráneas en las tablas.

El archivo generado se abrió en SQL Developer y se ejecutó, este empezó a agregar las tuplas en las tablas y logró velocidades de 10000 a 16000 tuplas por minuto al principio. Gráfico, Gráfico de líneas

Descripción generada automáticamente

Ilustración 39 Relación de datos cargados con los minutos transcurridos

Sin embargo, llegó un momento en el que por límites de poder de procesamiento disponible la velocidad bajó a 15 tuplas por minuto aun cuando la carga de datos se realizó desde una máquina virtual de la universidad para mayor velocidad. Podemos concluir que la perdida de velocidad se empezó a dar a partir de aproximadamente los 850000 datos.

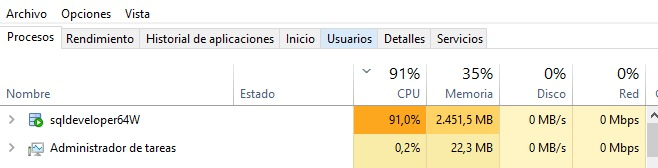


Ilustración 40 Uso del CPU que generó la carga de datos por SQL Developer junto con la memoria usada durante la carga

Durante la carga se decidió pausar el proceso de carga para empezar una carga de solo los datos faltantes y que estos no tardaran tanto, no obstante, al hacer esto el programa colapsó y dejó de responder antes de que se lograra hacer commit. Entonces se empezó a realizar la carga de un modo más segmentado, con alrededor de 300000 tuplas por ejecución en lugar de un millón.

Captura de pantalla de computadora

Descripción generada automáticamenteTexto

Descripción generada automáticamente con confianza bajaCon el método antes descrito se usó un archivo con 1081556 sentencias de insert y update que cargó en la base de datos un total de 1063558 tuplas.

Ilustración 41 Conteo de el total de tuplas existentes en la base de datos

Ilustración 42 Conteo de lineas en el archivo SQL generado

**Cambios interfaz**

En la interfaz se realizaron cambios en el tipo de usuario “Gerente general”, en el archivo de configuración de la interfaz para este tipo de usuarios se agregaron los 4 nuevos requerimientos de consulta ya que este usuario es quien realiza estas operaciones con el fin de hacer auditoría.

Texto

Descripción generada automáticamente

Ilustración 36 cambios en el archivo interfaceConfigAppGeneral.json

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Ilustración 43 nuevas opciones en el menú de la interfaz

En el archivo InterfazBancandesApp.java se creo una función por cada uno de los requerimientos de consulta y dos funciones adicionales que se encargan de convertir listas de VOClase a strings que se usan para actualizar la interfaz.

Para el primer requerimiento (consultar operaciones v2) la interfaz le muestra al usuario dos ventanas en las cuales el usuario ingresará las fechas inferior y superior respectivamente en formato dd/mm/aa, luego se le mostrará una ventana que le permitirá elegir el criterio por el cual desea filtrar la información, las ventanas siguientes dependen del criterio seleccionado, por cada criterio se creó una función en Bancandes.java, PersistenciaBancandes.java y SQLOperacion.java, como resultado se muestra en la interfaz las operaciones filtradas por el criterio elegido.

Para el segundo requerimiento (consultar operaciones v3) la interfaz le muestra al usuario dos ventanas en las cuales el usuario ingresará las fechas inferior y superior respectivamente en formato dd/mm/aa, luego se le mostrará una ventana que le permitirá elegir el criterio por el cual desea filtrar la información (recordar que en este caso el criterio se utiliza para tener en cuenta todas las operaciones que no lo cumplen), las ventanas siguientes dependen del criterio seleccionado, por cada criterio se creó una función en Bancandes.java, PersistenciaBancandes.java y SQLOperacion.java, como resultado se muestra en la interfaz las operaciones que no cumplen el criterio elegido.

Para el tercer requerimiento (consultar consignaciones) la interfaz le muestra al usuario dos ventanas en las cuales el usuario ingresará el valor y seleccionará el tipo de producto respectivamente. Para este requerimiento se creo una función en Bancandes.java, PersistenciaBancandes.java y SQLOperacion.java, como resultado se muestra en la interfaz las consignaciones obtenidas de la consulta.

Para el cuarto requerimiento (consultar punto(s) de atención) la interfaz le muestra al usuario dos ventanas en las cuales el usuario ingresará el id de cada puesto de atención. Para este requerimiento se crearon 2 funciones en los archivos Bancandes.java, PersistenciaBancandes.java y una función en SQLOperacion.java y SQLUsuario.java, esto se realizó de esta manera ya que en el requerimiento se pide mostrar los usuarios y también las operaciones hechas por estos usuarios, sin embargo en una sola consulta solo se obtendría un tabla con muchos datos que no pertenece a ninguna clase del negocio, por ello se decidió realizar dos consultas para poder obtener una lista de VOUsuario y una lista de VOOperacion y estas listas convertirlas en strings que se le enviará a la interfaz para actualizarse.

En la última línea del resultado de cada requerimiento se muestra un número que indica el tiempo en milisegundos que tardó la función que convierte la lista de VOClase a string, y en la consola se realiza un print que muestra el tiempo en milisegundos que se tardó la realización de la consulta más la actualización de la interfaz.

Por restricciones de hardware es imposible realizar consultas con una selectividad alta (más de 15.000 registros de respuesta) ya que la interfaz no responde y visual consume gran cantidad de recursos, tal vez en una maquina con mayor capacidad de ram se podrían hacer estas consultas. Debido a esto consideramos que llevar las tablas a memoria principal y luego realizar las operaciones por medio de operadores if, while, etc tardaría mucho más tiempo que en Oracle y consumiría gran cantidad de recursos.

Al comparar el rendimiento del requerimiento 10 en la interfaz visual y en SQL Developer podemos ver que la duración de las dos consultas en la interfaz es de en total 2 milisegundos, y para la mostrar los resultados tarda 333 milisegundos, mientras que realizar estas consultas en SQL Developer tarda 98 milisegundos para realizar las consultas y mostrar los resultados. Gracias a esto podemos definir que la interfaz visual es más eficaz para obtener la información y aplicarla en las clases definidas que sql para obtener los datos, sin embargo, la interfaz tarda mucho más en presentarlos al usuario.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteUna captura de pantalla de una computadora

Descripción generada automáticamente Captura de pantalla de computadora

Descripción generada automáticamente